

Description

ExpressCard with On-Card Flash Memory with Shared Flash-Control Bus but Separate Ready Lines

BACKGROUND OF INVENTION

[0001] This invention relates to flash-memory cards, and more particularly to ExpressCard flash cards with dual flash channels.

[0002] Flash memory is widely used for storing data in certain applications. Flash memory is especially useful for mobile and non-volatile applications, such as for portable or handheld devices. Flash memory is often more convenient than traditional mass storage devices such as hard disks. Flash memory also offers low power consumption, reliability, small size, and high speed.

[0003] Flash memory is non-volatile, since it retains stored data even after power is turned off. This is an improvement over standard random access memory (RAM), which is volatile and therefore loses stored data when power is

turned disconnected.

[0004] Universal–Serial–Bus (USB) is a widely used serial–interface standard for connecting external devices to a host such as a personal computer (PC). Another new standard is PCI Express, which is an extension of Peripheral Component Interconnect (PCI). An intent of PCI Express is to preserve and re–use PCI software.

[0005] As the number of mobile, portable, and handheld devices grows the popularity of flash memory increases. The most common type of flash memory is in the form of a removable memory card. This card allows the contents of the flash memory to be transferred easily between devices or computers.

[0006] However, when moving the flash memory card between devices, an additional host, reader, or adapter is often required for the host to communicate with the flash card. Many devices may not have the built–in ability to connect to a flash card, therefore a special adapter or card must be installed in the host device. In addition, the bus architecture can limit the speed of data transfer between the host and flash memory device.

[0007] Figures 1A–B show an ExpressCard. A new removable–card form–factor known as ExpressCard is being devel–

oped by the Personal-Computer Memory Card International Association (PCMCIA), PCI, and USB standards groups. ExpressCard 30 is about 75 mm long, 34 mm wide, and 5 mm thick and has ExpressCard connector 42, which fits a connector on a host when ExpressCard 30 is inserted into an ExpressCard slot on the host. The underside is shown in Fig. 1A while a top view is shown in Fig. 1B.

[0008] Figure 2 shows an enlarged version of ExpressCard. Some card applications may not fit in the small size of ExpressCard 30 of Figs. 1A-B, so an enlarged card size is also provided. ExpressCard 30' is also 75 mm long and has the same ExpressCard connector 42, but is wider (54 mm) at the opposite end from connector 42. The cutout notch from connector 42 to the wider part of the card is 22 mm deep. ExpressCard 30' is about 5 mm thick.

[0009] Figure 3 shows an ExpressCard interface to a host. A 26-pin connector is used to connect ExpressCard 30 to a host such as a PC. Power controller chip 34 receives power and ground supplies and various sensing and reset signals, and generates a Vcc power supply (such as 3.3 Volts) to ExpressCard 30. Other voltages such as 1.5 volts can be generated by power controller chip 34 and supplied to

ExpressCard 30. Multiple power and ground pins in the ExpressCard connector can improve signal quality and provide shielding. Clock and wake signals can also be provided to ExpressCard 30. Wake signal WAKE_REQ can be pulled high by a resistor on the host and pulled low by ExpressCard 30 to detect the presence of ExpressCard 30 in a slot on the host.

[0010] ExpressCard 30 can use a System-Management Bus (SMB) bus to transfer data to the host. Data and clock signals to and from ExpressCard 30 are coupled to SM bus controller 36. PCI Express data is transferred using the differential pair of PCI Express Transmit lines (PET) and the differential pair of PCI Express Receive lines (PER). Signal CPUSB# can be used for a CPU side-band.

[0011] ExpressCard 30 can also use USB to communicate with the host. Differential USB data signals USBD+ and USBD- are connected between ExpressCard 30 and host chip set 32. Host chip set 32 contains a USB host controller to facilitate communication with ExpressCard 30.

[0012] What is desired is an ExpressCard with flash-memory for data storage. An ExpressCard with an efficient flash-memory controller is desirable. An ExpressCard flash device that uses USB or PCI Express for communicating with

a host is desired.

BRIEF DESCRIPTION OF DRAWINGS

- [0013] Figures 1A–B show an ExpressCard.
- [0014] Figure 2 shows an enlarged version of ExpressCard.
- [0015] Figure 3 shows an ExpressCard interface to a host.
- [0016] Figure 4 is a block diagram of a flash–memory Express–Card.
- [0017] Figure 5 shows the flash–memory ExpressCard controller in more detail.
- [0018] Figure 6 shows two channels between the flash controller and the flash–memory chips on the ExpressCard.
- [0019] Figure 7 shows two channels of flash–memory chips with a shared control bus but separate ready lines.
- [0020] Figure 8 shows an embodiment using open–drain ready lines.
- [0021] Figure 9 shows an embodiment with four flash–memory chips in two channels.
- [0022] Figure 10 shows another embodiment with two independent channels and interleaving within each channel.

DETAILED DESCRIPTION

- [0023] The present invention relates to an improvement in flash

memory cards. The following description is presented to enable one of ordinary skill in the art to make and use the invention as provided in the context of a particular application and its requirements. Various modifications to the preferred embodiment will be apparent to those with skill in the art, and the general principles defined herein may be applied to other embodiments. Therefore, the present invention is not intended to be limited to the particular embodiments shown and described, but is to be accorded the widest scope consistent with the principles and novel features herein disclosed.

[0024] Figure 4 is a block diagram of a flash-memory Express-Card. ExpressCard connector 42 plugs into an Express-Card socket on a host, such as a PC, digital camera, PDA, music player, etc. While ExpressCard connector 42 has 26 electrical connector leads ("pins"), only four leads are needed for USB transfers – the differential USB data lines USB_{D+} and USB_{D-}, and power and ground.

[0025] Controller 40 connects to ExpressCard connector 42 over bus 44, which has the differential USB lines when controller 40 uses the USB protocol for host transfers. Other protocols, such as PCI Express, could use other signals in ExpressCard connector 42. Controller 40 acts as a USB

slave device, accepting and decoding commands from the host and responding to these commands, such as by transferring data or providing status information to the host.

[0026] Controller 40 can be a custom or semi-custom chip that contains all control functions for ExpressCard 30. Data from the host can be stored in flash-memory chips 38, 38', ... 38". Some ExpressCard 30 may have only one flash-memory chip 38 while others have multiple chips.

[0027] Flash bus 46 connects controller 40 to flash-memory chips 38, 38', 38". Flash bus 46 contains control signals and data signals, such as 8 bits of data. Commands and addresses can be sent as data over flash bus 46.

[0028] Figure 5 shows the flash-memory ExpressCard controller in more detail. Controller 40 can be a single-chip micro-controller based on microprocessor CPU 52, which can be an 8051 (8-bit), 80186 (16 bits), ARM CPU(32 bits), MIPS CPU(32/64 bits), etc. microprocessor core. Internal bus 66 connects CPU 52 with other blocks, such as read-only memory (ROM) 54, which can store program code executed by CPU 52, and RAM 56 which can be used by CPU 52 as a scratch-pad or parameter memory.

[0029] I/O control interface 58 can have I/O registers that drive

external pins of controller 40, and can be used to drive status LED's or detect when a write-protect switch is engaged. CPU 52 can write to these I/O registers to turn an LED on or off (or blink the LED) to indicate when a write to flash-memory on the ExpressCard is in progress.

[0030] Serial engine 50 contains logic to receive USB commands sent over the differential USB data lines from the host through the ExpressCard connector. The serial data is converted to parallel data words and stored in system buffer 64 or first in a FIFO memory in serial engine 50. Serial engine 50 controls the transfer of data to and from the ExpressCard connector over the USB data lines. When a command is detected on the USB data lines by serial engine 50, an interrupt to CPU 52 can be generated, allowing CPU 52 to read the command's data or parameters from serial engine 50 and perform the requested function.

[0031] CPU 52 can move data from serial engine 50 to system buffer 64, or can activate a direct-memory access (DMA) engine (not shown) to perform the transfer. System buffer 64 can act as a buffer, storing data from the host before it is written to the flash-memory chips. System buffer 64 can also act as a cache, storing data that was earlier read from the flash-memory chips by flash controller 60 and

making this data available more rapidly. Various read-ahead caching schemes can be implemented with the cache in system buffer 64.

[0032] Commands received from the host by serial engine 50 are decoded by CPU 52 and can include erase, write, and read commands for various sizes of data. CPU 52 performs these commands by sending addresses and internal high-level commands to flash controller 60, which contains state machines and counters to generate the proper low-level commands and timing required by the flash memory chips and perform these functions on blocks or pages of data in the flash memory chips. Flash controller 60 generates the necessary memory-control signals and chip commands such as chip selects, strobes, and read/write/erase commands, and keeps track of the current data byte being accessed or block begin erased. Memory mapping can be performed by CPU 52 to re-map pages of data and improve wear-leveling of memory locations in the flash-memory chips.

[0033] Some errors in the data stored in the flash memory chips can be corrected using error-correction code (ECC). As data is being written to the flash-memory chips, ECC generator 62 generates a multi-bit syndrome or ECC word to

append to the data. The data together with this ECC word are then sent to the flash-memory chips by flash controller 60 for storage. When the data is read back from the flash-memory chips by flash controller 60, this ECC word is stripped off the data and checked. When an error is detected, ECC generator 62 may correct the data word before the data is sent over internal bus 66 to system buffer 64. Alternatively, CPU 52 can be informed of ECC error details, and CPU 52 can correct the data before (or after) the data is sent to system buffer 64.

[0034] Figure 6 shows two channels between the flash controller and the flash-memory chips on the ExpressCard. Controller 40 is the primary controller chip on the ExpressCard and contains two flash controllers 60, 60', which generate external signals to the flash-memory chips. The flash-memory chips are arranged into two channels: flash-memory channel A 72 and flash-memory channel B 74.

[0035] The flash bus to the flash-memory chips from flash controllers 60, 60' can be divided into two separate channels. Data bus A 76 carries 8 bits of data to and from one or more flash-memory chips in flash-memory channel A 72, while data bus B 78 carries 8 bits of data to and from one

or more flash-memory chips in flash-memory channel B 74. Control signals in the flash bus are also divided into two channels. Control bus A 77 contains flash-chip-specific control signals for flash-memory channel A 72, while control bus B 79 contains flash-chip-specific control signals for flash-memory channel B 74. More channels could be added.

[0036] Flash-chip-specific control signals that can include chip-select, read and write enables, and address and command latch-enable signals. A write-protect signal may be tied to a fixed voltage and read by controller 40 through an I/O or input port.

[0037] Having separate channels to flash-memory chips allows for higher bandwidth transfers to and from the flash-memory chips, helping to improve the operating speed of the flash-memory ExpressCard. Dual flash channels and their higher data bandwidth are especially useful with higher-bandwidth protocols such as USB 2.0, since front and back end data rates are better matched.

[0038] Data stored to the two flash-memory channels could be interleaved, either at a low-level of one or more data bytes or at higher levels such as sectors, pages, or blocks. Alternate sectors, pages, or blocks are stored in alternat-

ing flash-memory channels to improve bandwidth. Erase operations could also be interleaved.

[0039] Figure 7 shows two channels of flash-memory chips with a shared control bus but separate ready lines. The flash-memory chips are arranged into two channels: flash-memory channel A 72' and flash-memory channel B 74', but more channels could be used. Controller 40' contains flash controller 60" that supports two or more flash-bus channels.

[0040] Most control signals in the flash bus are shared among the two channels. Control bus 80 contains most of the flash-chip-specific control signals for flash-memory channel A 72' and for flash-memory channel B 74'. When addresses and commands are sent through the data bus, the address or command values can be duplicated to both of data bus_A 76 and data bus_B 78.

[0041] Since flash-memory chips may differ in response times, such as the amount of time or delay to complete an erase, a write, or a read, the ready signal from different flash-memory chips may be generated at different times even when flash operations are started at the same time.

[0042] For example, a read operation to flash-memory chips in both channels 72', 74' may be initiated at the same time

by a command duplicated in both data buses and followed by a read-enable signal in control bus 80 that is shared and applied to both flash-memory chips in channels 72', 74' once the data is ready. However, the flash-memory chip being accessed in flash-memory channel A 72' is faster than the flash-memory chip being accessed in flash-memory channel B 74'. The channel A ready signal from the flash-memory chip in flash-memory channel A 72' is returned first on ready line 82. Later, perhaps several clock cycles later, the channel B ready signal from the flash-memory chip in flash-memory channel B 74' is returned on ready line 84.

[0043] Separate ready lines 82, 84 allow data to be transferred at a pace determined by the slower chip of the flash-memory chips. Data bus A 76 carries 8 bits of data to and from one or more flash-memory chips in flash-memory channel A 72', while data bus B 78 carries 8 bits of data to and from one or more flash-memory chips in flash-memory channel B 74'. Together the two bytes from the two flash channels can form a 16-bit data bus.

[0044] Having two channels allows for a larger page size and a wider data bus, increasing bandwidth.

[0045] Figure 8 shows an embodiment using open-drain ready

lines. Some flash-memory chips may have open-drain ready lines, allowing them to share the same ready line and assert ready at different times. Ready line 82' connects to both flash memory chips. Since shared control bus 80 and ready line 82' connect to both flash memory chips 102, 104, they act as one logical channel. For example, flash memory chip 102 can have upper bits 8–16 while flash memory chip 104 has lower bits 0–7.

[0046] Figure 9 shows an embodiment with four flash-memory chips in one logical channel with interleaving. Shared control bus 80 connects to all four flash memory chips 90, 92, 94, 96. However, Ready_1 88 connects to a first interleave of chips 90, 94 while Ready_2 89 connects to a second interleave of chips 92, 96. Chips 90, 94 are accessed together since they share chip-select CS0. Chips 92, 96 are accessed together since they share chip-select CS1. Flash-memory chips 90, 92 are in the upper portion of the data bus, or channel A_H, and receive data bus A, while flash-memory chips 94, 96 are in the lower portion of the data bus, or channel A_L, and receive data bus B. Ready_1 88 is driven by a first interleave of flash memory chips 90, 94, which are activated by chip-select CS0. Ready_2 89 is driven by a second interleave of flash memory chips 92,

96, which are activated by chip-select CS1. The interleaves thus include flash-memory chips in both upper and lower bits of the data bus. Interleaving can improve throughput since one interleaved chip can begin access while the other interleaved chips are finishing an access. For example, access can begin to the second interleaved chip 92, 96 while access is completing for the first interleaved chip 90, 94.

[0047] Figure 10 shows another embodiment with two independent channels and interleaving within each channel. Data bus A 76 and control bus A 80 connect to flash-memory chips 90, 92 in channel A. Data bus B 78 and control bus B 80' connect to flash-memory chips 94, 96 in channel B. Since separate control signals are applied to chips in each channel, the channels can be operated independently of each other.

[0048] Each flash-memory chip 90, 92, 94, 96 is controlled by its own dedicated chip-select signal CSA0, CSA1, CSB0, CSB1, respectively. Each flash-memory chip 90, 92, 94, 96 generates a separate ready signal Ready_A0, Ready_A1, Ready_B0, Ready_B1, respectively. Controller 60" can operate each channel independently of one another. Furthermore, operation and chips 90, 92 in channel A can be in-

terleaved by starting an operation or access to one chip 90 and then starting an operation or access to the other chip 92 before chip 90 has completed its operation. Likewise, operation or access of chips 94, 96 in channel B can be interleaved.

[0049] ALTERNATE EMBODIMENTS

[0050] Several other embodiments are contemplated by the inventor. For example controllers and functions can be implemented in a variety of ways. Functions can be programmed and executed by the CPU, or can be implemented in dedicated hardware, or in some combination. The ROM could be updateable, and some program code could be located in the RAM rather than the ROM. Some program code may be located in the flash memory chips and is uploaded to RAM when needed. Wider or narrower data buses and flash-memory chips could be substituted, such as 16 or 32-bit data channels. Alternate bus architectures with nested or segmented buses could be used internal or external to the controller. The ready line may be a busy or a not-busy line, and may be active high or low.

[0051] Rather than use USB for transfers, controller 40 of Fig. 4 could use other protocols, such as PCI Express, Firewire

(IEEE 1394), serial ATA, serial attached small-computer system interface (SCSI), etc. Different signals in the ExpressCard connector could be used for the different protocols with a different serial engine. For example, PCI Express can use the PET and PER signals in Fig. 3. A dual-mode controller could also be substituted for controller 40. Rather than have only a USB serial engine, a second serial engine could be added. The second serial engine connects to the PET and PER lines and follows the PCI Express protocol when communicating with the host through the ExpressCard connector.

[0052] Rather than have all flash-memory chips mounted directly on a board or other substrate in the ExpressCard, pairs of flash-memory chips can be stacked together in some embodiments. One flash-memory chip is directly put on top of another flash-memory chip. A very thin conducting material may be used for connections between the two flash-memory chips. The conventional flash-memory chip package has electrical signal leads (pins) and No Connect (NC) leads (pins). An NC pin has no electrical connection within the flash-memory chip package. All the respective electrical signals except the chip-select (CS) signal of each flash memory chip can share the same electrical lines. The

flash memory chips can be put on top of each other with all corresponding pins soldered to each other. However, the top chip's CS pin signal is re-routed to a NC lead on the bottom flash-memory chip and then to the substrate or printed-circuit board (PCB). Two or more flash chips can thus be stacked at one flash chip location on the board.

[0053] The abstract of the disclosure is provided to comply with the rules requiring an abstract, which will allow a searcher to quickly ascertain the subject matter of the technical disclosure of any patent issued from this disclosure. It is submitted with the understanding that it will not be used to interpret or limit the scope or meaning of the claims. 37 C.F.R. sect. 1.72(b). Any advantages and benefits described may not apply to all embodiments of the invention. When the word "means" is recited in a claim element, Applicant intends for the claim element to fall under 35 USC sect. 112, paragraph 6. Often a label of one or more words precedes the word "means". The word or words preceding the word "means" is a label intended to ease referencing of claims elements and is not intended to convey a structural limitation. Such means-plus-function claims are intended to cover not only the structures de-

scribed herein for performing the function and their structural equivalents, but also equivalent structures. For example, although a nail and a screw have different structures, they are equivalent structures since they both perform the function of fastening. Claims that do not use the word "means" are not intended to fall under 35 USC sect. 112, paragraph 6. Signals are typically electronic signals, but may be optical signals such as can be carried over a fiber optic line.

[0054] The foregoing description of the embodiments of the invention has been presented for the purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise form disclosed. Many modifications and variations are possible in light of the above teaching. It is intended that the scope of the invention be limited not by this detailed description, but rather by the claims appended hereto.